Besturingssystemen theorie

Academiejaar 2018 – 2019

HO GENT

Inhoudsopgave

HO GENT

Hoofdstuk 4: Processen

4.1. Wat is een proces?

Proces:
is een uitvoerbaar deel van een programma, dat in het geheugen geladen wordt en daar instructies doorgeeft naar de processor.

Multi:

- **multi-user** (veel gebruikers): verschillende gebruikers kunnen tezelfdertijd processen opstarten.
- **multi-tasking** (veel taken uitvoeren): een gebruiker kan meerdere processen tezelfdertijd opstarten.
- processen moeten beheerd worden: geheugen - schijfruimte - processor capaciteit - ….

HO GENT

4.2. Soorten processen

Er bestaan **drie soorten processen**:

1. **interactieve**
2. **automatische**
3. **daemons**

HO GENT

4.2. Soorten processen (2)

Interactieve processen:
opstarten en controleren vanuit een terminal sessie, m.a.w. er moet iemand aangemeld zijn op het systeem.

Interactieve processen kunnen zowel in de voorgrond (foreground) als in de achtergrond (background) draaien.

Foreground processen houden de terminal bezet zolang ze lopen.

Background processen bezetten de terminal niet en kunnen andere taken uitgevoerd worden.

HO GENT

4.2. Soorten processen (3)

Automatische (batch) processen:

Deze processen wachten eerst op uitvoering in een daartoe bestemde map. Vandaar uit worden ze opgeroepen door een programma dat de wachtrij analyseert en de programma's systematisch uitvoert.

Het programma dat het eerste in de wachtrij terecht kwam, wordt ook eerst uitgevoerd. De naam van dit systeem is "FIFO", wat staat voor "first in, first out".

Twee manieren:

1. **at** : zie later
2. **batch**: systeemadministratie - zie veel later (3^e jaar).

HO GENT

4.2. Soorten processen (4)

Daemons (demonen):

zijn server processen die continu draaien.

Meestal worden ze opgestart wanneer het systeem opstart, waarna ze in de achtergrond wachten tot hun diensten vereist zijn.

HO GENT

4.3. Background

Door middel van **job control** beheer je processen in foreground of background.

Een proces in background draaien heeft enkel zin als het over processen gaat die geen input verwachten en veel tijd nodig hebben.

Een proces opstarten in background: **commandonaam &**

PID: process identification - proces volgnummer.

Jobnumber - dit is een nummer dat door de shell gebruikt wordt.

HO GENT

4.4. Eigenschappen

Elk proces heeft een **aantal vaste eigenschappen**:

- Het **procesidentificatienummer of PID**: een uniek nummer dat gebruikt wordt om naar het proces te verwijzen.
- Het PID van het proces dat dit proces gestart heeft: parent process ID of **PPID**- letterlijk: het PID van de ouder.
- Het zogenaamde **nice number**: de mate van vriendelijkheid van dit proces: laat het nog veel processorcapaciteit over aan andere processen, of juist niet?
- De terminal van waaruit dit proces opgestart werd, als het om een interactief proces gaat. Dit wordt aangeduid met een **tty number**.
- De **gebruikersnaam** van de gebruiker aan wie het proces toebehoort.
- De **groepsnaam** van de groep aan wie het proces toebehoort.

HO GENT

4.5. Informatie weergeven

Korte info: **ps** zonder opties - process:

1. **PID**: het procesidentificatienummer.
2. **TTY**: het terminal type en nummer waaraan het proces verbonden is. Wij gebruiken pts, pseudo-terminals, in tegenstelling tot echte terminals waarbij je een toetsenbord en een scherm hebt, waarmee je niets anders kan doen dan 1 enkele shell openen, in een tekstuele omgeving (te vergelijken met DOS vroeger). Pseudo-terminals zijn terminal vensters in een grafische omgeving, of verbindingen vanop een netwerk.
3. **TIME**: een relatieve indicatie van de tijd die het aantal processorcycli dat het process al verbruikt heeft.
   Gewone processen van gebruikers verbruiken slechts een klein deel van de totale processorkracht.
4. **CMD**: de naam van het commando.

HO GENT

4.5. Informatie weergeven (2)

Uitgebreide info: **ps -ef**:

1. **UID**: de naam van de gebruiker die het proces opstartte.
2. **PID**: het procesidentificatienummer.
3. **PPID**: procesidentificatienummer van het *parent process, het proces dat dit proces opstartte.*
4. **TTY**: de terminal waaraan het proces verbonden is, "?" wil zeggen dat het proces niet aan een terminal verbonden is.
5. **CMD**: de naam van het commando.

HO GENT

4.5. Informatie weergeven (3)

top commando:

Geeft ongeveer dezelfde informatie als ps -ef, maar het wordt om de 5 seconden opgefrist.

Bovendien hebben we hier al automatisch een zeker vorm van sorteren: de zwaarste processen, dat wil zeggen de processen die het meeste processortijd verbruiken, worden bovenaan in de lijst getoond.

We krijgen ook niet alle processen te zien. Al naargelang de grootte van je terminal venster wordt de lijst ingekort. We krijgen dus een “top” van de processen te zien.

HO GENT

4.5. Informatie weergeven (4)

top commando:

De output van het **uptime** commando, met daarin informatie over hoe lang het systeem al draait, hoeveel gebruikers er verbonden zijn en wat de belasting is.

Het aantal processen en de status ervan: er draait altijd slechts 1 proces tegelijk op de CPU, terwijl de andere in een wachtrij staan.

De belasting van de processor(s): moet de processor veel berekeningen maken, dan is de belasting hoog.

Gebruik van het geheugen: alle programma's die actief zijn, nemen een plaatsje in op het geheugen.

Gebruik van de swap space (het virtuele geheugen): als er teveel programma's draaien, wordt alle beschikbare plaats in het geheugen opgevuld. Een speciale plek op de harde schijf wordt dan gebruikt als extra geheugen.

HO GENT

4.5. Informatie weergeven (5)

pstree commando:

samenhang van de processen

De meeste processen stammen af van **systemd**, het initiële proces waarmee het systeem gestart wordt.

HO GENT

4.6. Processen beheren

(deel van) Commando	Betekenis
commandonaam	Draait het commando in de voorgrond.
commandonaam &	Draait het commando in de achtergrond en geeft de terminal vrij.
jobs	Toon de commando's die in de achtergrond aan het draaien zijn.
Ctrl+Z	Bevries het commando (in het Engels: suspend).
Ctrl+C	Beëindig het commando dat in de voorgrond draait.
%n	Elk commando in de achtergrond krijgt een jobnummer (in bovenstaand voorbeeld: 1. Gebruik de uitdrukking % met dit nummer om naar een proces te verwijzen
bg	Aktiveer een bevroren commando terug, na Ctrl+Z.
fg	Breng een commando van de achtergrond naar de voorgrond.
kill	Beëindig een programma dat in de achtergrond draait.

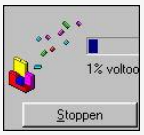
4.6. Processen beheren (2)

Enkele handige commando's:

- Het **time** commando werkt als een chronometer. Het geeft aan hoeveel uur, minuten en seconden een opdracht duurt om uit te voeren. Je gebruikt het door het te plaatsen vóór het commando dat je wilt uitvoeren.
- Het commando **uptime** geeft informatie over de belasting van het systeem.
- Het commando **w** geeft een overzicht van de aangemelde gebruikers en hun activiteit(en).

HO GENT

4.7. Processen beheren (3)



Indien één van de processen die je zelf hebt opgestart,
te veel middelen gebruikt, heb je twee mogelijkheden:

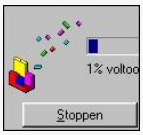
1. Zorg dat het proces minder belastend is door middel van het **renice** commando; hiervoor moet je het proces niet onderbreken.
2. Stop het proces.

De prioriteit veranderen

Het werken met **nice** en **renice** vereist een uitgebreide kennis van het systeem.
Er is echter een gemakkelijker manier: **top**.
Een belastend proces zal vermoedelijk een negatieve nice waarde hebben in de kolom "NI".

HO GENT

4.7. Processen beheren (4)



De prioriteit veranderen

Praktisch:

renice -n prioriteit -p PID

voorbeelden: zie labo's.

HO GENT

4.7. Processen onderbreken

Het **kill commando**

Een proces stoppen omdat het hangt, op hol slaat of teveel of te grote bestanden aanmaakt, doe je met **kill**.

Als je daartoe de gelegenheid hebt, probeer dan eerst de zachtaardige manier en stuur een **SIGTERM** (waarde 15) signaal.

Dit instrueert het proces om af te handelen waar het mee bezig is volgens de procedures zoals beschreven in de code van het programma. Op die manier wordt alle rommel opgeruimd en worden er geen bestanden beschadigd.

4.7. Processen onderbreken (1)

Het kill commando

- Zoek het procesidentificatienummer van het proces dat je wilt stoppen met behulp van **ps -ef.**
- Gebruik het commando **kill -15** ***PID_nummer.***
- Ga na met **ps** of het proces echt wel weg is.
- Van sommige processen raak je echter niet zo makkelijk af. Probeer dan in eerste instantie **kill -2,** een interruptiesignaal. Dat is hetzelfde als een programma onderbreken met Ctrl+C als het in de voorgrond draait.
- Als ook dat niet helpt, zit er niet veel anders op dan het sterkste signaal te sturen, een SIGKILL, met **kill -9.**
- Kijk in elk geval altijd na met **ps** of het stoppen gelukt is.

4.7. Processen onderbreken (2)

Het **xkill commando**

Grafische programma's die vasthangen kan je proberen stoppen met **xkill**.

Na het ingeven van dit commando verandert de muispijl in een doodshoofd. Beweeg het doodshoofd over het venster van het programma dat je wilt stoppen en klik met de linker muistoets.

4.8. Processen programmeren voor automatische uitvoering

Er zijn **drie manieren uitgestelde taken in te plannen**:

- Een korte tijd wachten en daarna de taak uitvoeren, gebruik makend van het **sleep** commando. Het tijdstip van uitvoering hangt af van het tijdstip waarop de taak gepland werd.
- De taak uitvoeren op een welbepaald tijdstip met het **at** commando. Het tijdstip van uitvoering is niet afhankelijk van het tijdstip van planning.
- Een taak steeds opnieuw uitvoeren, maandelijks, wekelijks, dagelijks, elk uur of elke minuut, door gebruik te maken van de **cron** faciliteit**.**

HO GENT

4.8. Processen programmeren voor automatische uitvoering (2)

Het **sleep** commando

Het enige dat sleep doet, is wachten.
Standaard wordt de wachttijd uitgedrukt in seconden.

HO GENT

4.8. Processen programmeren voor automatische uitvoering (3)

Het **at commando**

Geef het **at** commando in, gevolgd door het tijdstip waarop de geplande taak uitgevoerd moet worden.

Daarmee kom je in de at omgeving, gekenmerkt door de at prompt. Hier geef je het commando of de commando's in die gepland moeten worden.

Sluit af met **Ctrl+D** en de taak wordt gepland.

Je kan een overzicht krijgen van alle at jobs met het commando **atq**.
Met het **atrm commando** kan je de job verwijderen. Gebruik het **jobnummer** uit de eerste kolom van de output van atq
als argument.

HO GENT

4.8. Processen programmeren voor automatische uitvoering (4)

Het **cron systeem**:

De **cron daemon** draait constant op je systeem.

Deze dienst gaat elke minuut na of er taken uit te voeren zijn voor de gebruikers of voor de diensten die op een systeem draaien.

De taken worden opgeslagen in zogenaamde crontabs (tabellen).

Elke gebruiker kan een crontab hebben, waarin elke lijn een taak voorstelt die regelmatig herhaald moet worden.

HO GENT

4.8. Processen programmeren voor automatische uitvoering (5)

Het **cron systeem**:

Verder is er ook nog een crontab waarin de **systeem-specifieke taken** vernoemd worden, zoals bijvoorbeeld:

- Dagelijks de index maken waarvan het **locate** commando gebruik maakt;
- Dagelijks nagaan of er **updates** zijn voor de software op het systeem;
- Er dagelijks voor zorgen dat **logbestanden**, waarin informatie wordt opgeslagen over wat er allemaal op het systeem gebeurd is, niet te groot worden.
- Dagelijks en wekelijks een index maken van alle **man pagina's**, zodat apropos en whatis kunnen werken.

HO GENT

4.8. Processen programmeren voor automatische uitvoering (6)

Het **cron systeem**:

Andere taken kunnen zijn: het maken van backups, rapporten opmaken en doorsturen, systeeminformatie analyzeren en doormailen naar de administrator, herinneringsbrieven mailen, enzovoorts.

Alle taken die periodiek uitgevoerd moeten worden, komen voor opname in het cron systeem in aanmerking.

De crontabs voor het systeem vind je in de **/etc** map, die van de gebruikers in **/var/spool/cron/crontabs**, maar die map is niet toegankelijk voor de niet-geprivilegieerde gebruiker.
In **/var/spool/cron** vind je ook nog atjobs en atspool, omdat de at jobs onder de verantwoordelijkheid van de cron daemon vallen.

HO GENT

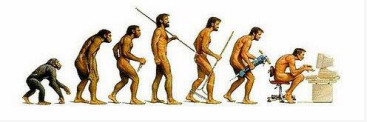
4.9. Levenscyclus

Een proces aanmaken

Een nieuw proces wordt aangemaakt doordat een bestaand proces een exacte kopie van zichzelf maakt.
Dit child process is eigenlijk net hetzelfde als het ouderproces, enkel het procesidentificatienummer verschilt.
Deze procedure heet men een **fork** (letterlijk: een vork of splitsing).

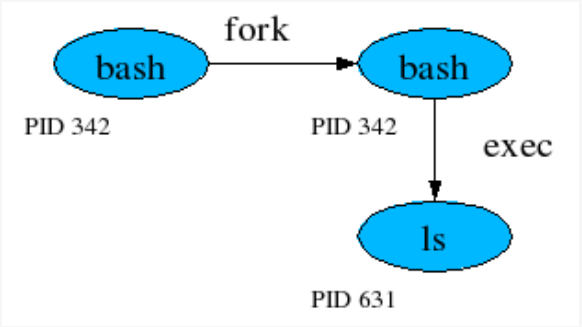
Na de fork wordt de geheugenruimte van het kindproces overschreven met de nieuwe procesdata: het commando dat gevraagd werd, wordt in het geheugen geladen.
Dit noemt men een **exec**.

Het geheel wordt **fork-and-exec** genoemd.



4.9. Levenscyclus (2)

Een proces aanmaken - fork-and-exec



HO GENT

4.9. Levenscyclus (3)

De rol van systemd

Zoals je kunt zien aan de output van het **pstree** commando, hebben veel processen systemd als ouderproces, terwijl dat helemaal niet mogelijk is.

Veel programma's "demoniseren" hun kindprocessen, zodanig dat die kunnen blijven draaien als de ouder stopt. Het systemd proces neemt de rol van peetvader van zulke processen: als de ouder sterft, vallen ze onder de verantwoordelijkheid van systemd.

Heel af en toe wil het nog wel eens mislopen met de "adoptie" van processen. Een proces dat geen ouderproces heeft, noemt men een zombie. Het systeem heeft geen vat meer op zo'n zombie-proces, het blijft in het geheugen hangen tot je de computer herstart.

HO GENT

4.9. Levenscyclus (4)

Een proces beëindigen

Wanneer een proces normaal eindigt, geeft het een code, de **exit status**, door aan de ouder. Als alles goed verlopen is, is de exit status nul.

De waarde van de exit status van shell commando's wordt opgeslagen in een speciale variabele, aangeduid met **$?**. Met het **echo** commando kan je de inhoud van deze variabele bekijken.

Processen eindigen omdat ze een signaal krijgen. Je kan verschillende signalen naar een proces sturen. Om dat te doen gebruik je het **kill commando**.

HO GENT